NEWTON-KRYLOV-SCHWARZ METHODS FOR RICHARDS' EQUATION *

E. W. JENKINS†, R. C. BERGER‡, J. P. HALLBERG‡, STACY E. HOWINGTON‡, C. T. KELLEY†, JOSEPH H. SCHMIDT‡, ALAN STAGG‡, AND M. D. TOCCI§

Abstract. In this paper we discuss the design and implementation of a Newton-Krylov-Schwarz solver for the implicit temporal integration on an unstructured three-dimensional spatial mesh of Richards' equation for groundwater flow in unsaturated porous media. We use aggregation techniques from the algebraic multigrid literature to construct a coarse mesh for two-level Schwarz methods. Our coarse mesh differs from other constructions in that no coarse mesh geometry need be created and we do not need geometric information about the subdomains. We report on a computational example to illustrate the performance of the preconditioner.

Key words. Newton-Krylov-Schwarz methods, Richards' equation, nonlinear equations, unstructured mesh

AMS subject classifications. 65H10, 65N55, 76S05, 76T05, 86A05

1. Introduction.

In this paper we discuss the design and implementation of a Newton-Krylov-Schwarz solver for the implicit temporal integration on an unstructured three-dimensional spatial mesh of Richards' equation for groundwater flow in unsaturated porous media. We use aggregation techniques from the algebraic multigrid literature [33] to construct a coarse mesh for two-level Schwarz methods. Our coarse mesh differs from other constructions, for example that in [9], in that no coarse mesh geometry need be created, and we do not need geometric information about the subdomains.

The mixed form of Richards' Equation is [8]

$$S_{\mathcal{S}} S(\psi) \frac{\partial \psi}{\partial t} + \eta \frac{\partial S(\psi)}{\partial t} = \nabla \cdot \left[K_{\mathcal{S}} k_r(\psi) \nabla (\psi + z) \right] + W \tag{1.1}$$

where ψ is pressure head; $S_{\mathcal{S}}$ is the specific storage, which accounts for water compressibility and aquifer elasticity; $S(\psi)$ is the water saturation or volumetric fraction of pore space occupied by water; η is the porosity or volumetric void fraction; K_S is the water-saturated hydraulic conductivity; k_r is the relative permeability of the media; and W is a source/sink term. In this formulation, the z axis is the vertical direction oriented positively upward. Both S and k_r are functions of ψ. K_S and $S_{\mathcal{S}}$ are provided as data.

The constitutive mode for saturation is

$$S = S_r + \frac{(1 - S_r)}{\left[1 + (\alpha |\psi|)^n\right]^m}, \quad \psi \leq 0 \tag{1.2}$$

*Version of October 13, 1999. This research was supported in part by Army Research Office contract DAAD19-99-1-0186, US Army contract DACA39-95-K-0098, National Science Foundation grant DMS-9700569, a Cray Research Corporation Fellowship, and a Department of Education GAANN fellowship. Computing activity was partially supported by an allocation from the North Carolina Supercomputing Center.

† North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA (ewjenkin@eos.ncsu.edu, Tim_Kelley@ncsu.edu,).

‡ US Army Waterways Experiment Station, Coastal And Hydraulics Laboratory and Information Technology Laboratory, 3909 Halls Ferry Road Vicksburg, Mississippi 39180, USA (schmidt@href.wes.army.mil, stagg@rusty.hpc.army.mil, stacy@hwy61.wes.army.mil, berger@hl.wes.army.mil, pettway@juanita.wes.army.mil,)

§ Department of Mathematical Sciences Worcester Polytechnic Institute Worcester, MA 01609, USA (mdtocci@wpi.edu)

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1999		**00-00-1999 to 00-00-1999**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Newton-Krylov-Schwarz Methods for Richards' Equation	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Waterways Experiment Station ,Coastal and Hydraulics Laboratory & Information Technology Laboratory,3909 Halls Ferry Road,Vicksburg,MS,39180	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**11**	

where S_r is the residual saturation, $\alpha > 0$ and $n > 1$ are parameters specific to the fluid and soil [25], and $m = 1 - 1/n$ [32]. The relative permeability function is [25]

$$k_r = \frac{\left[1 - (\alpha\,|\psi|)^{n-1}\,[1 + (\alpha\,|\psi|)^n]^{-m}\right]^2}{[1 + (\alpha\,|\psi|)^n]^{m/2}}, \quad \psi \le 0. \tag{1.3}$$

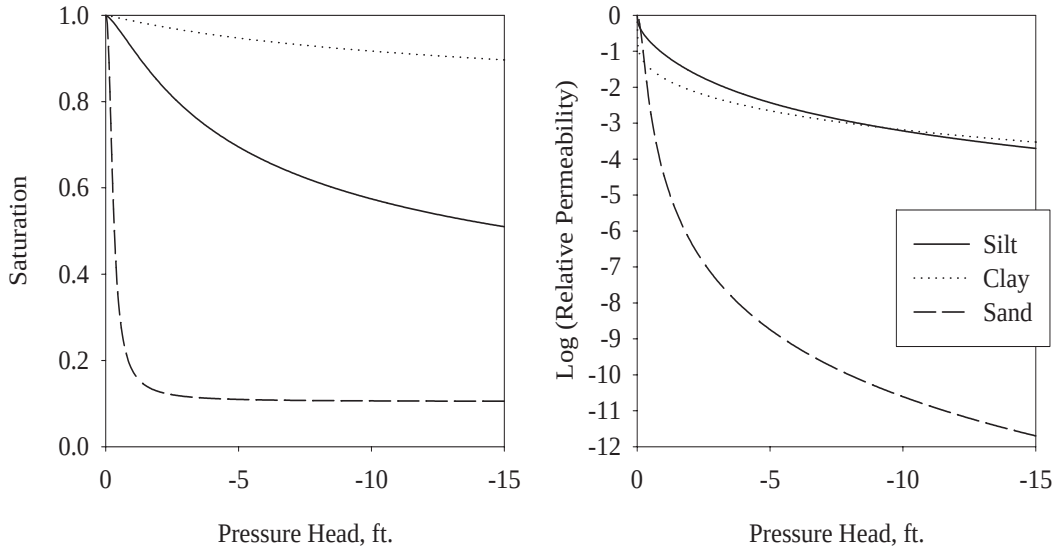
Both S and k_r are not differentiable at $\psi = 0$ when $1 < n < 2$. We address this nonsmoothness and the high cost of evaluation of the nonlinearities in (1.2) and (1.3) by approximating S and k_r with a spline [23, 29]. We used a piecewise linear spline in the computations reported here.

For the sand, silt, and clay soils used in the computations in § 3, we use parameters from the literature [7, 21]. These parameters are given in Table 1.1 and the saturation and relative permeability functions are plotted in Figure 1.1. Note the strong dependence of relative permeability on the pressure head in the Figure 1.1.

TABLE 1.1
Typical parameters for soil textural groups

Soil Textural Group	S_r (-)	α (1/ft)	n (-)
Sand	0.105	4.420	2.68
Silt	0.074	0.487	1.37
Clay	0.179	0.244	1.09

FIG. 1.1. *The ψ-S (left) and ψ-k_r (right) models used in the column drainage test.*



Implicit temporal integration in three space dimensions leads to large nonlinear equations at each time step. Newton-Krylov-Schwarz methods solve nonlinear equations by using Newton's method with a Schwarz domain decomposition preconditioned Krylov method to approximate the

Newton step. These methods have been used in computational fluid dynamics for some time [5, 6, 20]. This paper is one of the first to apply these methods to Richards' equation. Multigrid methods have also been applied to Richards' equation, [10, 28, 34, 35].

In § 2 we describe the nonlinear and linear solver issues raised by Richards' equation, introduce some notation from [27] for domain decomposition, and use that notation to describe our approach to Schwarz preconditioning.

In § 3 we report numerical results for the solution of a finite element discretization of Richards' equation in three space dimensions. Our numerical experiments, done on an IBM SP at the North Carolina Supercomputing Center (NCSC), show that the two-level preconditioners perform well and have very good scalability.

The work described in this paper is part of the development of the **Ad**aptive **H**ydrology model, or ADH. The ADH model is a production code being developed at WES and is designed for the analysis of basin scale hydrologic phenomena. The focus of this paper is the preconditioners and we use only fixed (*i. e.* time independent) unstructured meshes. We have also applied these ideas to surface water flow [17, 18]. We will describe adaptive spatial and temporal refinement and more complex simulations in future work.

2. Newton-Krylov-Schwarz methods. The weak formulation of Richards' equation leads to finite element discretizations that are implicit in time. An elliptic partial differential equation must be solved at each time step. After discretization one obtains a large system of nonlinear equations. In this section we begin with a description of inexact Newton methods [11] in general terms and then show how Newton-Krylov-Schwarz methods fit in that framework.

2.1. Nonlinear Solvers. We express nonlinear equations in the general form:

$$F(\xi) = 0. \tag{2.1}$$

We will assume that a solution ξ^* exists and that a good approximation to ξ^* is available. This latter assumption is appropriate in the context of implicit temporal integration, where ξ^* is the solution at the new time step and the initial approximation (the predictor) is an interpolation of solutions at previous times. We let $F'(\xi)$ denote the Jacobian of F at a vector ξ. We assume that $F'(\xi)$ is Lipshitz continuous in ξ and $F'(\xi^*)$ is nonsingular. These are standard assumptions in nonlinear equations [19] and are valid for the differential equations under consideration in this paper. The smoothness assumptions can be violated by the nonlinearities for Richards' equation in some cases [24], but this nonsmoothness can be managed by approximation with a well-chosen spline. Even the nonsmooth effects of a piecewise linear spline, which we use in our implementation, are benign.

We will describe Newton's method for (2.1) in the standard way [12, 19] in terms of the transition from a current approximation ξ_c to the solution ξ^* to a new approximation ξ_+ and describe the convergence in terms of the relative sizes of the new error $e_+ = \xi_+ - \xi^*$ and the current error $e_c = \xi_c - \xi^*$. The Newton iteration is

$$\xi_+ = \xi_c - F'(\xi_c)^{-1} F(\xi_c). \tag{2.2}$$

In (2.2) $F'(\xi_c)$ is the Jacobian matrix at the current iteration. Given sufficiently good data and sufficiently smooth nonlinearities the Newton iteration will converge quadratically, *i. e.*

$$\|e_+\| = O(\|e_c\|^2),$$

where $e = \xi - \xi^*$. In temporal integration applications the nonlinear iteration is terminated when

$$\|F(\xi)\| < \tau_a + \tau_r \|F(\xi_0)\|, \tag{2.3}$$

where τ_a and τ_r are absolute and relative error tolerances.

One does not compute the Newton step $s = -F'(\xi_c)^{-1}F(\xi_c)$ by using the inverse of the Jacobian but rather by solving (perhaps approximately) the linear equation

$$F'(\xi_c)s = -F(\xi_c). \tag{2.4}$$

For problems in three space dimensions the use of a direct method for solving (2.4) is impractical for reasons of both storage and computational cost. One must solve (2.4) by an iterative method. It is common [11, 19, 26] to terminate that linear iteration when the relative linear residual is small, *i. e.* when

$$\|F'(\xi_c)s + F(\xi_c)\| \leq \eta_r \|F(\xi_c)\| \tag{2.5}$$

for some small η_r. (2.5) is called the inexact Newton condition. The parameter η_r is called the forcing term. In applications to temporal integration, one can use absolute residuals and gain some efficiency [3]. We can express convergence results for both approaches in terms of the termination condition

$$\|F'(\xi_c)s + F(\xi_c)\| \leq \eta_a + \eta_r \|F(\xi_c)\|. \tag{2.6}$$

If the initial iterate is near the solution, the nonlinearity is sufficiently smooth, and (2.6) holds for some η_a and $0 \leq \eta_r < 1$ then the error in ξ_+ satisfies, [3, 11, 19],

$$\|e_+\| = O(\|e_c\|^2 + \eta_r \|e_c\| + \eta_a). \tag{2.7}$$

Clearly, if ξ_c is near ξ^* and η_r is sufficiently small the iteration will converge rapidly. However, solving the equation for the Newton step, (2.4), to very high accuracy may be wasteful, particularly if the initial iterate is far from the solution [13, 19].

2.2. The Linear Iteration and Preconditioning. If the linear equation (2.4) for the Newton step is solved by an iterative method, the overall iteration is called a Newton-Iterative method. The iteration (2.2) is called the *nonlinear iteration* or the *outer iteration*, and the iteration to solve the linear equation (2.4) is called the *linear iteration* or the *inner iteration*. If a Krylov method is used as the linear solver the combination was called a Newton-Krylov-Schwarz method in [5] and that term is now common. In this work we use the Krylov method BiCGSTAB [15, 19, 30] as the linear solver with an additive Schwarz preconditioner. Other low-storage transpose-free Krylov methods for nonsymmetric methods [14, 16] were tested in the early stages of this project along with BiCGSTAB(2), using the implementation from [31].

Preconditioning (from the right in our case), which is critical to good performance, replaces (2.4) with the equivalent system

$$F'(\xi_c)M\hat{s} = -F(\xi_c) \tag{2.8}$$

and then sets $s = M\hat{s}$. The preconditioning operator M is constructed so that (2.8) is easier to solve than (2.4).

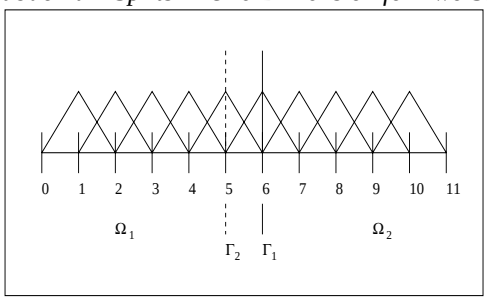
2.3. Schwarz Preconditioners. The discretized elliptic problems that must be solved at each time step have the general form

$$\mathcal{R}(U_h, v) = 0, \text{ for all } v \in \mathcal{V}^h \tag{2.9}$$

where $\mathcal{R}$ is the weak form of the nonlinear equation, $\mathcal{V}^h$ is the space of test functions and $U_h \in \mathcal{V}^h$. $\mathcal{R}$ is linear in the second argument and nonlinear in the first. $\mathcal{V}^h$ is the set of real-valued piecewise linear functions on the unstructured spatial mesh.

We begin by splitting the original physical domain Ω into subdomains Ω_i and restricting the action of the differential operator to the subdomains. The division leads to the creation of new boundary pieces, called artificial boundaries, for the subdomains. Figure 1.1 depicts a one–dimensional domain Ω that is split into two subdomains, Ω_1 and Ω_2. The artificial boundaries created by the split are Γ_1 and Γ_2, with Γ_1 part of the boundary for Ω_1 and Γ_2 part of the boundary for Ω_2.

FIG. 2.1. *Subdomain Splits in One-Dimension for Two Subdomains*



When creating the subdomains from the original domain, one can require that the subdomains share an artificial boundary or one can allow the subdomains to overlap, as depicted in Figure 2.1. Subdomains overlap if part or all of the artificial boundary for one subdomain lies in the interior of an adjacent subdomain. In Figure 2.1, the artificial boundary Γ_1 lies in the interior of the subdomain Ω_2, and similarly Γ_2 lies in Ω_1. In ADH, the domains overlap with an overlap width of h, the width of a single element. Increased overlap is difficult to achieve when using unstructured meshes and we have not attempted to incorporate it. The choice of overlapping subdomains led to the use of Schwarz-type preconditioners. General discussions of Schwarz preconditioners may be found in [27].

The closure of the differential operator restricted to the subdomains requires that boundary conditions be placed on the artificial boundaries, as these boundaries were not part of the original problem formulation. Schwarz methods use zero Dirichlet boundary conditions on the artificial boundaries [27] because the algorithms incorporate error corrections on the artificial boundaries during the subdomain solves.

The one-level additive Schwarz preconditioner is a block Jacobi preconditioner. Let A be the discretization of the differential operator, and assume that the number of nodes of the discrete problem included in Ω_i is n_i. Define the matrix R_i to be the (discrete) restriction operator for subdomain i, i.e., $R_i = \begin{bmatrix} 0 & I & 0 \end{bmatrix}$ where I is of size $n_i \times n_i$. If

$$B_i = R_i^T \left(R_i A R_i^T \right)^{-1} R_i$$

then the one-level additive Schwarz preconditioner is

$$M = \sum_{i=1}^{p} B_i \tag{2.10}$$

where p is the number of subdomains.

This preconditioner is implemented readily on a multiprocessor computer by assigning one or more subdomains to a processor. Because the overlap between subdomains is minimal and and zero Dirichlet conditions are imposed on the artificial boundaries, there is no need for communication after the application of the one–level additive Schwarz preconditioner.

We found that the performance of the one-level method is inferior to two-level methods, which we describe now. A two-level method requires a coarse mesh for either the entire computational domain or a coarse resolution of the entire domain.

Similarly to the one level preconditioner, we define a coarse mesh restriction operator R_0 and let $B_0 = R_0^T \left(R_0 A R_0^T \right)^{-1} R_0$. The two-level additive Schwarz preconditioner is

$$M = B_0 + \sum_{i=1}^{p} B_i \tag{2.11}$$

and the two-level hybrid II preconditioner [22, 27] is given by

$$M = B_0 + (I - B_0 A) \sum_{i=1}^{p} B_i. \tag{2.12}$$

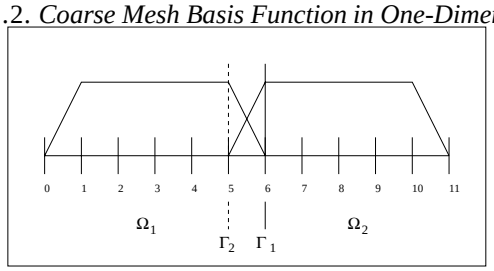
In the work reported here we found that the alternative form

$$M = B_0 + \sum_{i=1}^{p} B_i \left(I - A B_0 \right) \tag{2.13}$$

performed better.

To avoid generating and storing a separate coarse mesh, we define coarse mesh basis functions as aggregates of already existing fine mesh basis functions, an idea from algebraic multigrid [33]. One-dimensional examples of these coarse mesh basis functions are depicted in Figure 2.2, with one coarse mesh basis function defined per subdomain.

FIG. 2.2. *Coarse Mesh Basis Function in One-Dimension*



The coarse mesh basis functions are formed by summing the fine mesh functions having support in a given subdomain. If $\{v_i\}$ is the nodal basis for $\mathcal{V}^h$ and $\{D_I\}$ is the set of nodes in subdomain I, then the coarse mesh functions V_I are formed by

$$V_I = \sum_{i \in D_I} v_i.$$

The coarse mesh inverse B_0 can be constructed from the fine mesh Jacobian in a direct way.

$$B_0 = R_0^T \left[R_0 J \left(U_h, v_i \right) R_0^T \right]^{-1} R_0$$

where J is the Jacobian of $\mathcal{R}$ and R_i is defined by its action on a vector, i.e.,

$$\left(R_i u \right)_I = \sum_{i \in D_I} u_i .$$

2.4. Parallel implementation. We assign one or more subdomains to each processor. Elements along the interprocessor boundaries are shared by those processors owning any of its nodes. The nodal information for the boundary elements must be communicated among processors so that each processor sees all of the shared elements.

The linear solver without preconditioning requires two types of communication: (1) an update of the nodal values along processor edges for each matrix vector product and (2) a global sum for each inner product. The additive Schwarz / block Jacobi preconditioning does not require any communication. The coarse mesh preconditioning also requires two types of communication. Initially, each processor forms part of the coarse matrix, and a global communication is used to assemble these parts on all of the processors. The coarse mesh problem is then solved redundantly on every processor. Each application of the preconditioner also requires the communication of the residual vector. Each processor sums its pieces of the residual vector, and a global communication is used to pass these pieces to all of the processors. The coarse preconditioner is then applied to the reduced residual vector. Each processor then expands the appropriate portion of the reduced vector and updates its portion of the full residual vector.

3. Numerical Results. In this section we report on a simulation of a heterogeneous column experiment. The physical problem, pictured in Figure 3.1, is a column filled with a mixture of clay, silt, and sand. The column is primarily sand, with a clay lens near the bottom of the column and silt lenses in several places throughout the column. Boundary conditions applied to the column were prescribed total head (pressure head plus elevation from a datum) at the bottom and no-water-flux boundaries on the sides and top. The no flow condition imposed on the top of the column means that there is no flow from the water phase through the top of the column. Air is assumed to be present everywhere and will enter the column onasce the water is removed from the pore spaces. Initially, the column is water saturated. At the start of the simulation, the bottom boundary condition is gradually modified to atmospheric pressure head and the column was allowed to drain by gravity for 100 seconds.

FIG. 3.1. *3D Heterogeneous Column*

We discretized the equation on an unstructured tetrehedral mesh. We used the piecewise constant in time and piecewise linear in space finite element discretizations from [1]. The residual formulation of Richards' equation is

$$
\begin{aligned}
R\quad & (\psi, w) = \textstyle\int_{Q_n} R(\psi)\, w^h \, dQ \\
& = \textstyle\int_{Q_n} \left\{ S_{\mathcal{S}} S(\psi) \frac{\partial \psi^h}{\partial t} + \eta \frac{\partial S\left(\psi^h\right)}{\partial t} \right\} w^h \, dQ - \int_{Q_n} \nabla w^h : \left[K_{\mathcal{S}} k_r \left(\psi^h\right) \nabla \left(\psi^h + z\right) \right] dQ \\
& - \textstyle\int_{\Omega_n} \left(w^h\right)_n^+ \cdot \left\{ S_{\mathcal{S}} \left[\left(S\left(\psi^h\right) \psi^h \right)_n^+ - \left(S\left(\psi^h\right) \psi^h \right)_n^- \right] + \eta \left[S\left(\psi^h\right)_n^+ - S\left(\psi^h\right)_n^- \right] \right\} d\Omega \\
& - \textstyle\int_{\hat{P}_n} \left[w^h \cdot K_{\mathcal{S}} k_r \left(\psi^h\right) \nabla \left(\psi^h + z\right) \right] \cdot \hat{n} \, dP + \int_{Q_n} w^h \cdot W \, dQ
\end{aligned}
\tag{3.1}
$$

In (3.1) Ω is the column, Γ the boundary, $Q_n = \Omega \times (t_n, t_{n+1}]$, and $P_n = \Gamma \times (t_n, t_{n+1}]$. $\hat{P}_n$ denotes the part of P_n where Neumann conditions are applied.

We report on two discretizations of the column, one roughly 8 times the size of the other. In this way h, the typical cell diameter for the mesh, and H, the typical subdomain size, are both approximately halved as the mesh is refined. This allows us to gauge the scalability of the iteration. The coarse mesh was generated automatically and refined by hand, (roughly) halving the mesh width in each of the x, y, and z directions. The small mesh has 5881 nodes and 30720 elements and the large mesh has 43889 nodes and 245760 elements. The meshes were generated using the Groundwater Modeling System (GMS) [2] and subdomains were constructed using the node ordering from GMS.

We terminate the linear iteration using (2.6), the l^∞ norm. and $[\eta_a, \eta_r] = [10^{-7}, 0]$. We terminate the nonlinear iteration when (2.3) holds with $[\tau_a, \tau_r] = [10^{-5}, 0]$. In the experiments reported here only one nonlinear iteration was needed at each time step.

Updating preconditioning information infrequently in a temporal integration is common practice. In codes like DASPK, [4], for example, the decision to update preconditioning information at a given time step can be tied to the performance of the linear or nonlinear solver. A two-level preconditioner has two parts, the subdomain solvers and the coarse mesh work, *i. e.* computation and factorization of the coarse mesh matrix. In this computation we found that the coarse mesh work scaled poorly. The reasons for this are that each processor must compute a part of the coarse mesh matrix and communicate it to the others and that a dense matrix LU factorization was used because we had no way, other than brute force, to obtain sparsity information about the coarse mesh matrix. We reduced computation time by 20% by not performing the coarse mesh work at every time step. In the computations reported here, the coarse mesh work was done every 10 nonlinear iterations.

A more sophisticated strategy for updating both components of the preconditioner is needed and will be the subject of future research.

In Table 3.1 we report on the number of linear iterations for the entire simulation (LIC) for four preconditioners: point Jacobi, one- and two-level additive Schwarz, and the two-level hybrid II preconditioner given by (2.13). We give the time in seconds for the entire simulation (TS) and the cumulative time for the linear solver (TL). The final row in the table is the ratio of iteration counts for the two problems, which we use as a measure of scalability.

Table 3.1 shows that the one-level preconditioner is worse than point Jacobi in terms of computational time and that both the two-level and the hybrid two-level preconditioner are far better in terms of both number of iterations and computer time.

TABLE 3.1
3D Heterogeneous Column Results

Small: 5 PEs, 8 blocks				
	Point Jacobi	One-Level	Two-Level	Hybrid
LI	13364	8292	2137	1493
TS	84	133	74	71
TL	50	99	32	30
Large: 40 PEs, 8 blocks				
LI	19394	16783	2688	1848
TS	184	275	107	106
TL	130	235	63	55
Ratio	1.45	2.02	1.26	1.24

The computations reported here were performed on an IBM SP with 177 200Mhz Power3 processors on 2-way SMP nodes with 1Gb of memory per node running IBM AIX Version 4.3, IBM PE 2.4, and IBM C for AIX Version 4.4.

The results of the test problems show that the incorporation of a coarse mesh problem into the preconditioner is necessary to obtain a significant reduction in the computational time. The aggregate elements appear to have been a sensible choice for the coarse mesh problem and were easy to implement.

Acknowledgments. The authors wish to thank David Keyes, Casey Miller, and Jun Zou for many helpful discussions. The authors also wish to thank Bob Walkup and Dave Klepacki of IBM Research and Eric Sills and Mark Reed of the North Carolina Supercomputing Center for their help in optimizing and running the ADH Model on the IBM SP.

REFERENCES

[1] M. BEHR AND T. E. TEZDUYAR, *Finite element solution strategies for large–scale flow simulations*, Computer Methods in Applied Mechanics and Engineering, 112 (1994), pp. 3–24.

[2] BRIGHAM YOUNG UNIVERSITY, *GMS – The Department of Defense Groundwater Modeling System Reference Manual*, Brigham Young University Engineering Computer Graphics Laboratory, Provo, Utah, 1994.

[3] P. N. BROWN AND A. C. HINDMARSH, *Matrix-free methods for stiff systems of ODE's*, SIAM J. Numer. Anal., 23 (1986), pp. 610–638.

[4] P. N. BROWN, A. C. HINDMARSH, AND L. R. PETZOLD, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[5] X.-C. CAI, W. D. GROPP, D. E. KEYES, AND M. D. TIDRIRI, *Newton-Krylov-Schwarz methods in CFD*, in Proceedings of the International Workshop on the Navier-Stokes Equations, R. Rannacher, ed., Notes in Numerical Fluid Mechanics, Braunschweig, 1994, Vieweg Verlag.

[6] X.-C. CAI, D. E. KEYES, AND V. VENKATAKRISHNAN, *Newton-Krylov-Schwarz: an implicit solver for CFD*, in Domain Decomposition Methods in Sciences and Engineering, Proceedings of the 8th International Conference on Domain Decomposition Methods, R. Glowinski, J. Periaux, Z. Shi, and O. Widlund, eds., Chichester, 1997, Wiley, pp. 387–402.

[7] R. F. CARSEL AND R. S. PARRISH, *Developing joint probability distributions of soil water retention characteristics*, Water Resources Research, 24 (1988), pp. 755–769.

[8] M. A. CELIA, E. T. BOULOUTAS, AND R. L. ZARBA, *A general mass-conservative numerical solution for the unsaturated flow equation*, Water Resources Research, 26 (1990), pp. 1483–1496.

[9] T. F. CHAN, S. GO, AND J. ZOU, *Boundary treatments for multilevel methods on unstructured grids*, SIAM J. Sci. Comput., 21 (1999), pp. 46–66.

[10] C. DAWSON, H. KLIE, C. S. WOODWARD, AND M. F. WHEELER, *A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver*, Computational Geosciences, 1 (1997), pp. 215–249.

[11] R. DEMBO, S. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[12] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[13] S. C. EISENSTAT AND H. F. WALKER, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[14] R. W. FREUND, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.

[15] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, no. 17 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1997.

[16] M. H. GUTKNECHT, *Variants of BICGSTAB for matrices with complex spectrum*, SIAM J. Sci. Comput., 14 (1993), pp. 1020–1033.

[17] S. E. HOWINGTON, R. C. BERGER, J. P. HALLBERG, J. F. PETERS, A. K. STAGG, E. W. JENKINS, AND C. T. KELLEY, *A model to simulate the interaction between groundwater and surface water*, Tech. Rep. CRSC-TR99-27, North Carolina State University, Center for Research in Scientific Computation, August 1999. Proceedings of the High Performance Computing Users' Group Meeting, Monterrey, CA, June 7–10.

[18] E. W. JENKINS, R. C. BERGER, J. P. HALLBERG, S. E. HOWINGTON, C. T. KELLEY, J. H. SCHMIDT, A. K. STAGG, AND M. D. TOCCI, *A two-level aggretation-based Newton-Krylov-Schwarz method for hydrology*. To appear in the Proceedings of the International Parallel CFD Conference, Williamsburg, VA, May 23–26, 1999.

[19] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[20] D. E. KEYES, *Aerodynamic applications of Newton-Krylov-Schwarz solvers.* to appear in Proc. of 14th International Conference on Num. Meths. in Fluid Dynamics. (R. Narishima et al. eds), Springer, NY 1995.

[21] F. J. LEIJ, W. J. ALVES, M. T. VAN GENUCHTEN, AND J. R. WILLIAMS, *The UNSODA Unsaturated Soil Hydraulic Database, User's Manual, Version 1*, US Environmental Protection Agency, Washington, DC, 1996. EPA/600/R-96/095.

[22] J. MANDEL, *Balancing domain decomposition*, Communications in Applied Numerical Methods 9, (1993), pp. 233–241.

[23] C. T. MILLER AND C. T. KELLEY, *A comparison of strongly convergent solution schemes for sharp front infiltration problems*, in Computational Methods in Water Resources X, Vol. 1, A. Peters, G. Wittum, B. Herrling, U. Meissner, C. Brebbia, W. Gray, and G. Pinder, eds., Kluwer Academic Publishers, 1994, pp. 325–332.

[24] C. T. MILLER, G. A. WILLIAMS, C. T. KELLEY, AND M. D. TOCCI, *Robust solution of Richards' equation for non-uniform porous media*, Water Resources Research, 34 (1998), pp. 2599–2610.

[25] Y. MUALEM, *A new model for predicting the hydraulic conductivity of unsaturated porous media*, Water Resources Research, 12 (1976), pp. 513–522.

[26] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[27] B. SMITH, P. BJØRSTAD, AND W. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, 1996.

[28] C. A. S. SOUCIE, *Mixed Finite Element Methods for Variably Saturated Subsurface Flow*, PhD thesis, Rice University, 1996.

[29] M. D. TOCCI, C. T. KELLEY, AND C. T. MILLER, *Accurate and economical solution of the pressure head form of Richards' equation by the method of lines*, Advances in Water Resources, 20 (1997), pp. 1–14.

[30] H. A. VAN DER VORST, *Bi–CGSTAB: A fast and smoothly converging variant to Bi–CG for the solution of nonsymmetric systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[31] ———, *Parallel iterative solution methods for linear systems arising ¿from discretized PDEs*, 1995. in *Special Course on Parallel Computing in CFD, AGARD-R-807* (AGARD, Neuilly-sur-Seine).

[32] M. T. VAN GENUCHTEN, *Predicting the hydraulic conductivity of unsaturated soils*, Soil Science Society of Americal Journal, 44 (1980), pp. 892–898.

[33] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.

[34] C. S. WOODWARD, *A Newton-Krylov-multigrid solver for variably saturated flow problems*, Tech. Rep. UCRL-JC-129371, Lawrence Livermore National Laboratory, January 1998. Submitted to XII International Conference on Computational Methods In Water Resources.

[35] C. S. WOODWARD AND J. E. JONES, *Parallel Newton-Krylov-multigrid solvers for variably saturated flow problems*. Presentation at the Fifth SIAM Conference on Mathematical and Computational Issues in the Geosciences, San Antonio, TX, March, 1999.